



## สิเมอร์ช (Simurgh)

ตามตำนานของชาวเปอร์เซียโบราณในซาร์ฮามะ ชาลวีร์บุรุษในตำนานแห่งเปอร์เซียได้ตกหลุมรักกับยูดาบาซึ่งเป็นเจ้าหญิงแห่งคาบูล ชาลได้ขอเจ้าหญิงแต่งงาน แต่พระบิดาของเจ้าหญิงได้ให้ปริศนาเพื่อทดสอบปัญญาของชาลดังนี้

ในเปอร์เซียมีเมืองอยู่จำนวน  $n$  เมืองกำกับด้วยหมายเลข  $0$  ถึง  $n - 1$  และมีถนนไปกลับ (two-way roads) จำนวน  $m$  เส้นกำกับด้วยเลข  $0$  ถึง  $m - 1$  ถนนแต่ละเส้นเชื่อมระหว่างสองเมืองที่แตกต่างกัน และแต่ละคู่ของเมืองจะมีถนนอย่างมากไม่เกินหนึ่งเส้น มีถนนบางเส้นถูกกำหนดให้เป็น *ถนนหลวง* สำหรับการเดินทางของราชวงศ์เท่านั้น ปัญหาของชาลคือต้องระบุให้ได้ว่าถนนเส้นใดบ้างที่เป็นถนนหลวง

ชาลมีแผนที่ระบุเมืองและถนนทั้งหมดในเปอร์เซีย แต่เขาไม่รู้ว่าถนนเส้นใดบ้างที่เป็นถนนหลวง ทั้งนี้เขาสามารถที่จะขอความช่วยเหลือจากสิเมอร์ชซึ่งเป็นนกกแห่งความเมตตาในตำนานและเป็นผู้ปกป้องของชาล อย่างไรก็ตามสิเมอร์ชไม่ต้องการที่จะบอกชาลตรง ๆ ว่าถนนเส้นใดเป็นถนนหลวง สิเมอร์ชจึงบอกชาลเพียงว่าเซตของถนนหลวงทั้งหมดก็คือ *โกลเด้นเซต* (golden set) โดยเซตของถนนใด ๆ จะเป็นโกลเด้นเซตก็ต่อเมื่อ

- มีถนนจำนวน *เท่ากับ*  $n - 1$  เส้นและ
- ในแต่ละคู่ของเมือง การจะไปถึงเมืองหนึ่งจากอีกเมืองหนึ่งสามารถใช้เพียงแค่นถนนในเซตนี้ได้

มากกว่านั้น ชาลสามารถถามสิเมอร์ชได้บางคำถาม การถามแต่ละครั้งจะมีขั้นตอนดังนี้

1. ชาลเลือก *โกลเด้นเซต* ของถนนขึ้นมา และจากนั้น
2. สิเมอร์ชจะบอกชาลว่า มีถนนหลวงจำนวนเท่าใดในโกลเด้นเซตนั้น

โปรแกรมของคุณต้องช่วยชาลหาเซตของถนนหลวง ด้วยการถามสิเมอร์ชได้มากที่สุด  $q$  คำถาม โดยเกรดเดอร์จะทำตัวเป็นสิเมอร์ชให้

## รายละเอียดการเขียนโปรแกรม

คุณต้องพัฒนาโปรแกรมต่อไปนี้

```
int[] find_roads(int n, int[] u, int[] v)
```

- $n$  คือ จำนวนเมือง
- $u$  และ  $v$  เป็นอาร์เรย์ความยาว  $m$  สำหรับทุก  $0 \leq i \leq m - 1$  โดย  $u[i]$  และ  $v[i]$  เป็นเมืองที่ถูกเชื่อมด้วยถนนหมายเลข  $i$
- โปรแกรมจะต้องคืนค่าอาร์เรย์ความยาว  $n - 1$  ที่ระบุหมายเลขของถนนหลวง (เรียงลำดับอย่างไรก็ได้)

คำตอบของคุณสามารถเรียกใช้โปรแกรมย่อยในเกรดเดอรัต่อไปนี้ได้มากที่สุด  $q$  ครั้ง

```
int count_common_roads(int[] r)
```

- $r$  อาร์เรย์ความยาว  $n - 1$  ระบุหมายเลขของถนนในโกลเด้นเซต (เรียงลำดับอย่างไรก็ได้)
- โปรแกรมจะคืนค่าตอบเป็นจำนวนของถนนหลวงใน  $r$

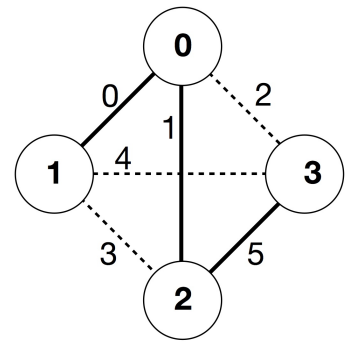
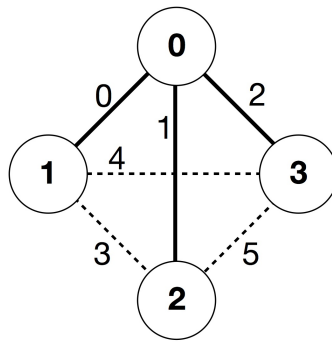
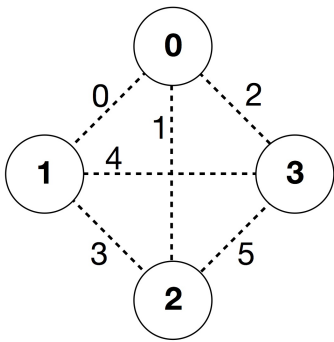
## ตัวอย่าง

```
find_roads(4, [0, 0, 0, 1, 1, 2], [1, 2, 3, 2, 3, 3])
```

find\_roads(...)

count\_common\_roads([0, 1, 2]) = 2

count\_common\_roads([5, 1, 0]) = 3



ในตัวอย่างนี้มีเมืองจำนวน 4 เมืองและถนนจำนวน 6 เส้น กำหนดให้  $(a, b)$  คือถนนที่เชื่อมระหว่างเมือง  $a$  และเมือง  $b$  โดยถนนถูกกำกับด้วยหมายเลข 0 ถึง 5 โดยมีลำดับดังต่อไปนี้  $(0, 1)$ ,  $(0, 2)$ ,  $(0, 3)$ ,  $(1, 2)$ ,  $(1, 3)$ , and  $(2, 3)$  ทุกๆโกลเด้นเซตมีจำนวนถนน  $n - 1 = 3$  เส้น

กำหนดให้ถนนหลวงถูกกำกับด้วยเลข 0 1 และ 5 นั่นคือถนน  $(0, 1)$ ,  $(0, 2)$ , และ  $(2, 3)$  ดังนั้น

- `count_common_roads([0, 1, 2])` จะคืนค่า 2 เพราะการถามครั้งนี้เกี่ยวกับถนนหมายเลข 0, 1, และ 2 นั่นก็คือ ถนน  $(0, 1)$   $(0, 2)$  และ  $(0, 3)$  ซึ่งมีถนนสองเส้นที่เป็นถนนหลวง
- `count_common_roads([5, 1, 0])` จะคืนค่า 3 คำตอบนี้หมายถึงเซตของถนนหลวงทั้งหมด

โปรแกรม `find_roads` จะคืนค่าตอบ `[5, 1, 0]` หรืออาร์เรย์อื่นใดที่มีความยาว 3 ที่บรรจุถนนทั้งสามเส้นนี้

สังเกตว่า การเรียกใช้ต่อไปนี้ไม่ได้รับอนุญาตให้ใช้

- `count_common_roads([0, 1])`: ความยาวของ  $r$  ไม่เท่ากับ 3
- `count_common_roads([0, 1, 3])` ในที่นี้  $r$  ไม่สามารถเป็นโกลเด้นเซตได้ เนื่องจากไม่สามารถเดินทางจากเมือง 0 ถึง 3 ได้ด้วยถนน  $(0, 1)$ ,  $(0, 2)$ ,  $(1, 2)$  เพียงเท่านั้น

## ข้อจำกัด

- $2 \leq n \leq 500$

- $n - 1 \leq m \leq n(n - 1)/2$
- $0 \leq u[i], v[i] \leq n - 1$  (สำหรับทุก  $0 \leq i \leq m - 1$ )
- สำหรับทุก  $0 \leq i \leq m - 1$ , ถนน  $i$  เชื่อมสองเมืองที่ต่างกัน (นั่นคือ  $u[i] \neq v[i]$ ).
- ระหว่างสองเมืองใด ๆ มีถนนได้อย่างมากหนึ่งเส้น
- สองเมืองใด ๆ จะสามารถเดินทางถึงกันได้จากถนนเหล่านี้
- เซตของถนนหลวงเป็นโกลเด้นเซต
- `find_roads` สามารถเรียก `count_common_roads` ได้อย่างมาก  $q$  ครั้ง ในการเรียกแต่ละครั้ง เซตของถนนที่ระบุด้วย  $r$  ต้องเป็นโกลเด้นเซต

## ปัญหาย่อย

1. (13 คะแนน)  $n \leq 7, q = 30\,000$
2. (17 คะแนน)  $n \leq 50, q = 30\,000$
3. (21 คะแนน)  $n \leq 240, q = 30\,000$
4. (19 คะแนน)  $q = 12\,000$  และมีถนนเชื่อมทุกคู่ของเมือง
5. (30 คะแนน)  $q = 8000$

## เกรตเตอร์ตัวอย่าง

เกรตเตอร์ตัวอย่างอ่านข้อมูลนำเข้าด้วยรูปแบบต่อไปนี้

- บรรทัดที่ 1:  $n\ m$
- บรรทัดที่  $2 + i$  (สำหรับทุก  $0 \leq i \leq m - 1$ ):  $u[i]\ v[i]$
- บรรทัดที่  $2 + m$ :  $s[0]\ s[1]\ \dots\ s[n - 2]$

ในที่นี้  $s[0], s[1], \dots, s[n - 2]$  เป็นหมายเลขกำกับถนนหลวง

เกรตเตอร์ตัวอย่างให้ผลลัพธ์ YES ถ้า `find_roads` เรียก `count_common_roads` อย่างมาก 30 000 ครั้ง และคืนค่าคำตอบเซตของถนนหลวงที่ถูกต้อง ไม่เช่นนั้นผลลัพธ์คือ NO

พึงตระหนักว่า `count_common_roads` ในเกรตเตอร์ตัวอย่างไม่ได้ตรวจสอบว่า  $r$  มีสมบัติเป็นโกลเด้นเซตหรือไม่ แต่กระนั้นเกรตเตอร์ตัวอย่างจะนับจำนวนและคืนค่าจำนวนถนนหลวงในอาร์เรย์  $r$  อย่างไรก็ดี ในระบบตรวจจริง ถ้าโปรแกรมของคุณเรียกใช้ `count_common_roads` โดยที่ไม่ใช่โกลเด้นเซต เกรตเตอร์จริงจะให้คำตอบเป็น 'Wrong Answer'.

## ข้อสังเกตทางเทคนิค

โปรแกรม `count_common_roads` ในภาษา C++ และภาษา Pascal ใช้วิธีการ *pass by reference* ด้วยเหตุผลเชิงประสิทธิภาพ คุณสามารถเรียกใช้โปรแกรมได้เป็นปกติ เกรตเตอร์จริงในระบบตรวจจะรับประกันว่า ค่าของ  $r$  จะไม่มีการเปลี่ยนแปลง